REMARKS

The title was objected to as not being descriptive. Applicants have amended the title and respectfully submit that the title is descriptive.

Claims 1-26 stand rejected under 35 USC § 102(e) as being anticipated by US Pat. No. 6,535,905 (hereafter Kalafatis). Applicants respectfully submit that applicants' claims are distinguishable over Kalafatis.

In view of the Examiner's read of applicants' claims to cover a "branch" instruction, applicant has clarified that claim 1 covers an instruction that "direct[s] said processor to suspend execution of said first thread". This distinguishes the "first instruction" from a "branch" instruction which might, as a side effect, cause a thread switch. Notably, Kalafatis suggests embodiments where some branches do have this side effect (Col. 10, ll. 53-56) and others where all branches do (Col. 10, ll. 56-59). Thus, the branch itself is not an instruction that directs the processor to suspend execution of the thread. It is a branch which directs the processor to branch and may or may not have the side effect of causing a thread *switch*.

In view of the Examiner's interpretation, applicant has also submitted Intel's US 6,496,925, which discusses other instances of events that not only cause a thread *switch*, but also may cause thread exit (e.g., "Halt" instruction). See Col. 24, l. 55, et seq. ("Thread Exit and Entry") and Col. 23, l. 59 – Col. 23, l. 55 ("The Active Thread State Machine"). Applicant hereby stipulates, only for the purpose of simplifying prosecution of this application, that these portions of US 6,496,925 also describe a product that was on sale for more than one year before the filing date of the present application.

Applicant submits, however, that applicants' claims are patentable over US 6,496,925 and the Examiner's cited art. As discussed above, the branch instruction of Kalafatis is not an instruction directing suspension of execution of a thread (nor is a halt, which halts execution). Moreover, Kalafatis does not teach de-partitioning or annealing resources in response to the various thread switching events such as branch instructions.

While Kalafatis discusses thread *switching*, the discussion of partitioning is limited to the fact that the resources are partitioned in the multithreaded mode. See, e.g., Fig. 4 and discussion at Col. 7, ll. 6 – 61:

> The logical partitioning for two threads of the buffering (or storage) facilities of a functional unit may be achieved by allocating a first predetermined set of entries within a buffering resource to a first thread and allocating a second predetermined set of entries within the buffering resource to a second thread. Specifically, this may be achieved by providing two pairs of read and write pointers, a first pair of read and write pointers being associated with a first thread and a second pair of read and write pointers being associated with a second thread. The first set of read and write pointers may be limited to a first predetermined number of entries within a buffering resource, while the second set of read and write pointers may be limited to a second predetermined number of entries within the same buffering resource.

Kalafatis does not suggest that thread switching should also cause the de-partitioning of resources. The mere switching to another thread does not automatically cause the prior thread's partitioned resources to be released for use by the other thread. Indeed, if any thread switch could cause de-partitioning of resources, this could, for example, lead to very inefficient program execution because partitioning and de-partitioning of processor resources may in some cases be a time consuming task.

For example, thread switching may be done in t he case where a thread is very active. See "The Forced Thread Change Logic (160)", Kalafatis, Col. 16, l. 51 – Col. 18, l. 40). To relinquish partitions of resources for an overactive thread may be inefficient and is certainly not taught or suggested by Kalafatis.

Applicant now claims in claim 1 that the processor includes a plurality of partitionable resources. In response to a program instruction that "direct[s] the processor to suspend execution of" the first thread, the processor is (1) "to suspend execution of the first thread" and (2) actually "relinquish portions of ... partitionable resources ..." These three aspects independently form grounds to distinguish Kalafatis.

Applicants submit that all claims now pending are in condition for allowance. Such action is earnestly solicited at the earliest possible date. If there is a deficiency in fees, please charge our Deposit Acct. No. 02-2666.

Respectfully submitted,

Date: 1/3/05

Jeffrey S. Draeger, Reg. No. 41,000

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8598

FIRST CLASS CERTIFICATE OF MAILING
(37 C.F.R. § 1.8(a))

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

on January 3, 2005.
Date of Deposit

Conny Willesen
Name of Person Mailing Correspondence

Conny Willesen — 01-03-2005
Signature — Date